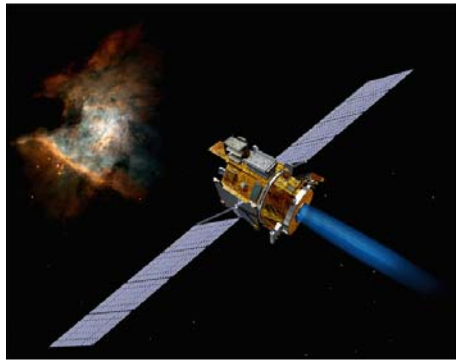
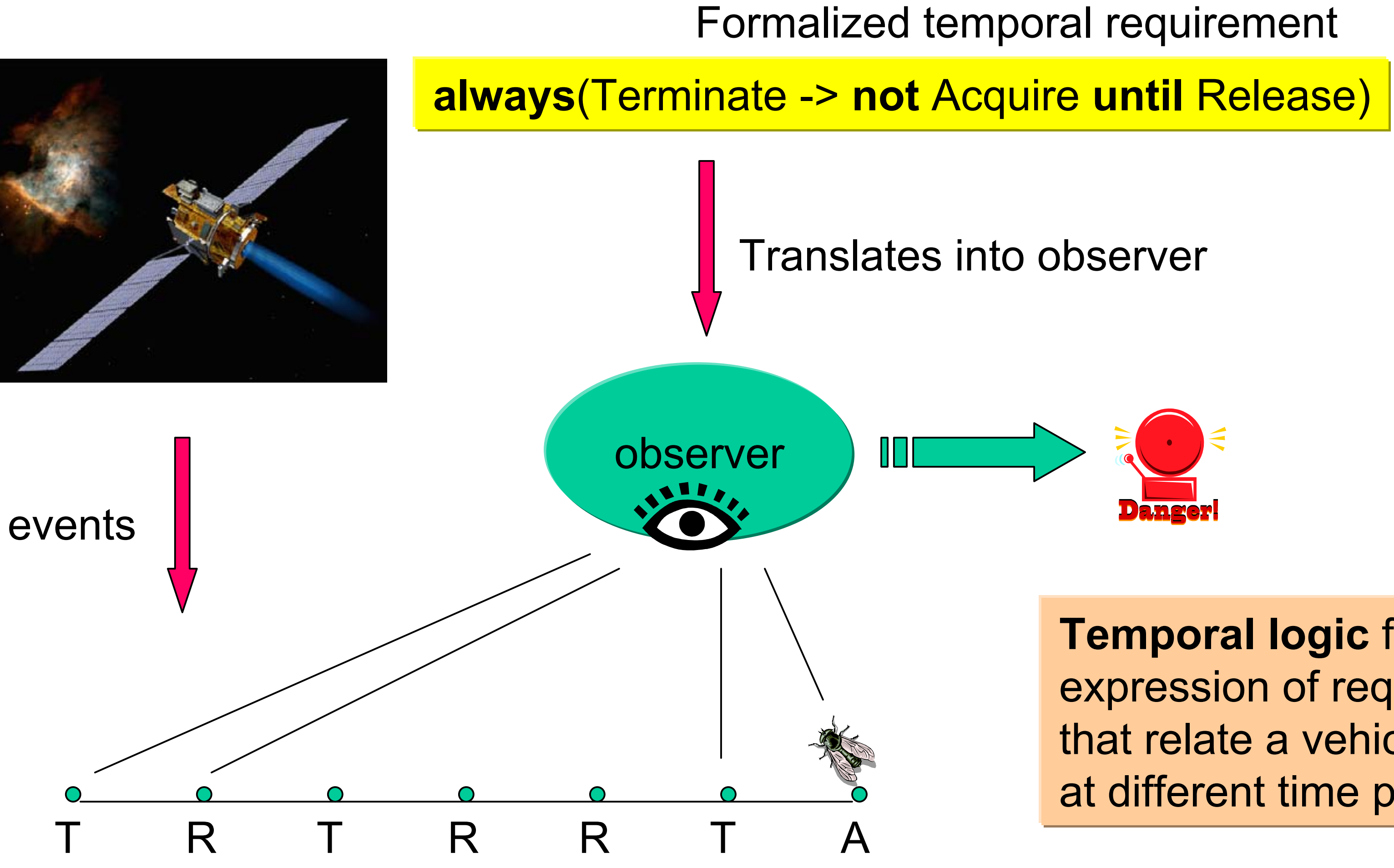
Information Sciences & Technology
Ames Research Center

Explanation of Accomplishment

- **POC:** Klaus Havelund (ASE group, Code IC, havelund@email.arc.nasa.gov)
- **Background:** Klaus Havelund and Grigore Rosu (former ASE, now at University of Illinois at Urbana-Champaign) took the initiative to organize the 1st International workshop on Runtime Verification RV'01 (http://ase.arc.nasa.gov/rv), with peer-reviewed papers, which was held as a satellite event to the prestigious CAV conference (Computer Aided Verification). Runtime Verification represents highly scalable techniques for analyzing the execution of instrumented programs, with the purpose of detecting errors. Havelund and Rosu were invited to submit the 5 best ranked papers to a special issue of the Formal Methods in Systems Design journal. Havelund and Rosu's paper *"An Overview of the Runtime Verification Tool Java PathExplorer"* were among the selected papers (reviewed independently by the PC). The RV workshops continue, with RV'02 (http://ase.arc.nasa.gov/rv2002) already held, also organized by Havelund and Rosu, and RV'03 being planned in connection with CAV'03 in Colorado.
- **Accomplishment:** The Java PathExplorer tool provides a set of algorithms for instrumenting and monitoring Java programs. A program is instrumented to emit events when executed. The event stream makes up an execution trace, which is then analyzed by the different algorithms. The tool has new, efficient algorithms for detecting deadlocks and data races, and violations of temporal logic formulas.
- **Future work:** We are currently extending the capability of JPaX to be able to detect other kinds of concurrency errors. Currently we are implementing an algorithm for detecting higher level data races. Errors in the Remote Agent found by the POC were caused by such higher level data races. We are also extending the tool to find other forms of deadlocks, also referred to as communication deadlocks. Current activities focus on the development of a more powerful temporal logic, in which real-time constraints can be formulated, and in which data values can be referred to. This work will find application in developing a testing environment for the K9 rover.

Runtime Verification with Java PathExplorer





Formalized temporal requirement
always(Terminate -> **not** Acquire **until** Release)
Translates into observer
observer
Danger!
events
T
R
T
R
R
T
A

Temporal logic facilitates expression of requirements that relate a vehicle's states at different time points.

Java PathExplorer is a software tool for efficiently monitoring the execution of a program, checking that the execution trace satisfies a set of formalized temporal requirements.